L Number	Hits	Search Text	DB	Time stamp
1	248	(RST or restart adj2 (marker or flag)) with (head or beginning or start) with line	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 13:47
2	1242915	compression or encoding or coding	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:30
3	2565	(compression or encoding or coding) with still adj1 image	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:30
4	3290	(compression or encoding or coding) same still adj1 image	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:32
5	1	((RST or restart adj2 (marker or flag)) with (head or beginning or start) with line) and (((compression or encoding or coding) same still adj1 image) or JPEG)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:31
6	48	((RST or restart adj2 (marker or flag)) with (head or beginning or start) with line) and (compression or encoding or coding)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:32
7	1	((RST or restart adj2 (marker or flag)) with (head or beginning or start) with line) and JPEG	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:33
8	28242	intra-frame or intraframe or i-frame or i adj1 frame	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:34
9	7	((RST or restart adj2 (marker or flag)) with (head or beginning or start) with line) and (intra-frame or intraframe or i-frame or i adj1 frame)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:34
10	2482	(RST or restart adj2 (marker or flag)) with (head or beginning or start)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:58

11	8288	(compression or encoding or coding) and (intra-frame or intraframe or i-frame or i adj1 frame)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:39
12	23	((RST or restart adj2 (marker or flag)) with (head or beginning or start)) and JPEG	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:58
13	9	((RST or restart adj2 (marker or flag)) with (head or beginning or start)) and ((compression or encoding or coding) and (intra-frame or intraframe or i-frame or i adj1 frame))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:52
15	3	(insert$4 or plac$4 or add or adding) adj2 (RST or restart adj2 (marker or flag)) with (head or beginning or start) with (line or row)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:57
16	3	(insert$4 or plac$4 or add or adding) adj2 (RST$2 or restart adj2 (marker or flag)) with (head or beginning or start) with (line or row)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:57
17	3	(insert$4 or plac$4 or add or adding) adj2 (RST$2 or restart adj2 (marker or flag or bit)) with (head or beginning or start) with (line or row)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:58
18	2742	(RST$2 or restart adj2 (marker or flag)) with (head or beginning or start)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 13:30
19	24	((RST$2 or restart adj2 (marker or flag)) with (head or beginning or start)) and JPEG	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:59
20	1	(((RST$2 or restart adj2 (marker or flag)) with (head or beginning or start)) and JPEG) not (((RST or restart adj2 (marker or flag)) with (head or beginning or start)) and JPEG)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 10:59
21	15	(insert$4 or plac$4 or add or adding) adj3 (RST$2 or restart adj2 (marker or flag)) with (head or beginning or start)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:21
22	13	("4300206" | "4493049" | "5239625" | "5257113" | "5329616" | "5634089" | "5758042" | "5801716" | "5828814" | "5850504" | "5884014" | "5949438" | "5953007").PN.	USPAT	2004/06/24 13:47
23	21	6049390.URPN.	USPAT	2004/06/24 13:47

24	2	((RST or restart adj2 (marker or flag)) with (head or beginning or start)) and ((("4300206" | "4493049" | "5239625" | "5257113" | "5329616" | "5634089" | "5758042" | "5801716" | "5828814" | "5850504" | "5884014" | "5949438" | "5953007").PN.) or 6049390.URPN.)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 13:50
25	37	((RST or restart adj2 (marker or flag)) with (head or beginning or start)) and MCU	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 13:50
26	2	(insert$4 or plac$4 or add or adding) adj3 (RST$2 or restart adj2 (marker or flag)) with correlat$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:22
27	2	(insert$4 or plac$4 or add or adding) adj3 (RST$2 or restart) with correlat$6	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:23
28	31	(insert$4 or plac$4 or add or adding) adj3 (RST$2 or restart) with (head or beginning or start)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:26
29	708	(RST$2 or restart) with (head or beginning or start) with (line or row)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:28
31	161	((RST$2 or restart) with (head or beginning or start) with (line or row)) and (compression or encoding or coding)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:29
32	113965	(compression or encoding or coding) same (image or picture)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:29
30	4	((RST$2 or restart) with (head or beginning or start) with (line or row)) and JPEG	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:29
33	24	((RST$2 or restart) with (head or beginning or start) with (line or row)) and ((compression or encoding or coding) same (image or picture))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 14:31

L Number	Hits	Search Text	DB	Time stamp
1	7	(determin$4 or calculat$6) adj3 (restart adj2 interval)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
2	13	("4635258" | "4703481" | "4740969" | "5138617" | "5193186" | "5500809" | "5513315" | "5537539" | "5715386" | "5748882" | "5784607" | "5862308" | "5905857").PN.	USPAT	2004/06/24 15:18
3	4	6453430.URPN.	USPAT	2004/06/24 15:18
4	0	((("4635258" | "4703481" | "4740969" | "5138617" | "5193186" | "5500809" | "5513315" | "5537539" | "5715386" | "5748882" | "5784607" | "5862308" | "5905857").PN.) or 6453430.URPN.) and (restart adj2 interval)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:19
5	14	((("4635258" | "4703481" | "4740969" | "5138617" | "5193186" | "5500809" | "5513315" | "5537539" | "5715386" | "5748882" | "5784607" | "5862308" | "5905857").PN.) or 6453430.URPN.) and (restart or RST)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:19
6	84518	(compression or encoding or coding) with (image or picture)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
7	0	(((("4635258" | "4703481" | "4740969" | "5138617" | "5193186" | "5500809" | "5513315" | "5537539" | "5715386" | "5748882" | "5784607" | "5862308" | "5905857").PN.) or 6453430.URPN.) and (restart or RST)) and (((compression or encoding or coding) with (image or picture)) or JPEG)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
8	113965	(compression or encoding or coding) same (image or picture)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
9	0	(((("4635258" | "4703481" | "4740969" | "5138617" | "5193186" | "5500809" | "5513315" | "5537539" | "5715386" | "5748882" | "5784607" | "5862308" | "5905857").PN.) or 6453430.URPN.) and (restart or RST)) and ((compression or encoding or coding) same (image or picture))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
10	456	(restart adj2 interval)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
11	85	((compression or encoding or coding) same (image or picture)) and ((restart adj2 interval))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:21
12	80	((restart adj2 interval)) and JPEG	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:44

13	5	(((compression or encoding or coding) same (image or picture)) and ((restart adj2 interval))) not (((restart adj2 interval)) and JPEG)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:44

L Number	Hits	Search Text	DB	Time stamp
1	14	(defin$4 or comput$6 or decid$4) adj3 (restart adj2 interval)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 15:48

L Number	Hits	Search Text	DB	Time stamp
1	89	(RST or restart adj2 (marker or flag)) with (left) with (line or row)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 16:02
2	132	(RST or restart) with (left) with (line or row)	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 16:02
3	43	((RST or restart) with (left) with (line or row)) not ((RST or restart adj2 (marker or flag)) with (left) with (line or row))	USPAT; US-PGPUB; EPO; JPO; DERWENT; IBM_TDB; USOCR	2004/06/24 16:02

Your search matched **2** of **1046194** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevanc Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

(restart interval) Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

---

1 **Detection and correction of transmission errors in JPEG images**
Yi-Huang Han; Jin-Jang Leou;
Circuits and Systems for Video Technology, IEEE Transactions on , Volume: 8 , Issue: 2 , April 1998
Pages:221 - 231

[Abstract] [PDF Full-Text (300 KB)] **IEEE JNL**

---

2 **Restart marker regulation technique for progressive JPEG image co in mobile communications**
Tien-Hsu Lee; Hsiu-Hua Hsu; Pao-Chi Chang;
Communications Letters, IEEE , Volume: 4 , Issue: 12 , Dec. 2000
Pages:411 - 413

[Abstract] [PDF Full-Text (64 KB)] **IEEE JNL**

Your search matched **1** of **1046194** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevance Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

restart <sentence> (start <or> beginning <or> head [Search]

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

1 **Improved power quality with "written-pole" motor-generators and "written-pole" motors**
Roesel, J.F., Jr.;
Southeastcon '96. 'Bringing Together Education, Science and Technology'., Proceedings of the IEEE , 11-14 April 1996
Pages:565 - 570

[Abstract] [PDF Full-Text (480 KB)] **IEEE CNF**

Your search matched **0** of **1046194** documents.
A maximum of **500** results are displayed, **15** to a page, sorted by **Relevanc** **Descending** order.

Refine This Search:
You may refine your search by editing the current search expression or enteri new one in the text box.

rst <sentence> (start <or> beginning <or> head <or Search

☐ Check to search within this result set

Results Key:
JNL = Journal or Magazine **CNF** = Conference **STD** = Standard

Results:
No documents matched your query.

14

Detection and Correction of Transmission Errors in JPEG Images

Yi-Huang Han and Jin-Jang Leou

***Abstract*—In this study, the detection and correction approach to transmission errors in Joint Photographic Experts Group (JPEG) images using the sequential discrete cosine transform (DCT)-based mode of operation is proposed. The objective of the proposed approach is to eliminate transmission errors in JPEG images. Here a transmission error may be either a single-bit error or a burst error containing N successive error bits. For an entropy-coded JPEG image, a single transmission error in a codeword will not only affect the underlying codeword, but may also affect subsequent codewords. Consequently, a single error in an entropy-coded system may result in a significant degradation. To cope with the synchronization problem, in the proposed approach the restart capability of JPEG images is enabled, i.e., the eight unique restart markers (synchronization codewords) are periodically inserted into the JPEG compressed image bitstream. Transmission errors in a JPEG image are sequentially detected both when the JPEG image is under decoding and after the JPEG image has been decoded. When a transmission error or equivalently a corrupted restart interval is detected, the proposed error correction approach simply performs a sequence of bit inversions and redecoding operations on the corrupted restart interval and selects the "best" feasible redecoding solution by using a proposed cost function for error correction. Based on the simulation results obtained in this study, the proposed approach can recover high-quality JPEG images from the corresponding corrupted JPEG images at bit error rates (BER's) up to approximately 0.4%. This shows the feasibility of the proposed approach.**

***Index Terms*—Bit inversion, burst error, detection and correction, JPEG image, transmission error.**

I. Introduction

To reduce transmission bit rate or storage capacity, many compression techniques have been developed for a broad spectrum of applications, such as videoconferencing, videophones, facsimile transmission, and high-definition TV [1], [2]. Standardization of image compression techniques has become an important issue for interoperability of equipment from different manufacturers. The most important international compression standards for facsimile (binary) images, still images, videoconferencing (low-resolution moving pictures and associated audio) and HDTV (high-resolution moving pictures and associated audio) are Groups 3, 4, and JBIG (Joint Bilevel Image Experts Group) [3], [4], JPEG (Joint Photographic Experts Group) [5], [6], H.261 [7], and H.263, and MPEG (Motion Picture Experts Group) [8], respectively. In this study, JPEG, the compression standard for still images, will be treated.

For an entropy-coded JPEG image, a single transmission error in a codeword will not only affect the underlying codeword, but may also affect subsequent codewords, resulting in a great degradation of the received image. As an illustrative example shown in Fig. 1(b), the quality of a corrupted JPEG image without a synchronization codeword is indeed poor. To cope with the synchronization problem, synchronization codewords can be periodically inserted into the JPEG compressed image bitstream so that synchronization can be achieved once any synchronization codeword is correctly received. In JPEG, if the restart capability is enabled, the eight unique restart markers provided by JPEG can be used as eight synchronization codewords. However, as an illustrative example shown in Fig. 1(c), even the restart capability is enabled, the quality of a corrupted JPEG image is still greatly degraded. To improve the quality of the corrupted JPEG images, in this study the detection and correction approach to transmission errors in JPEG images is proposed.

To protect image quality against transmission errors in compressed images, there are several proposed approaches [8]–[26], including: 1) the channel coding approach; 2) the error resilience approach; and 3) the detection and correction approach. For the channel coding approach, some error correction codes are used to encode images such that transmission errors can be detected and corrected (partially or completely) by adding redundancy to the transmitted information [9]. However, the channel coding approach will moderately increase the transmission bit rate.

In ISO/IEC Recommendation H.262 (ISO/IEC 13818-2) [8], several error resilience techniques, namely, error concealment, spatial localization, and temporal localization, have been addressed. For block-based image/video compression standards, such as JPEG, H.261, H.263, and MPEG (H.262), several error concealment methods have been proposed, which can be divided into three categories, namely, spatial error concealment methods [10]–[13], temporal error concealment methods [14], [15], and hybrid concealment methods [16]–[18]. Additionally, three simple error concealment methods are often employed by block-based compressed images using DCT (discrete cosine transform): 1) zero-substitution, which simply replaces all pixels in a corrupted block by zeros; 2) mean-substitution, which uses the mean of the external boundary samples of

Manuscript received August 8, 1995; revised September 16, 1997. This paper was recommended by Associate Editor J. Woods. This work was supported in part by the National Science Council, R.O.C., under Grant NSC84-2213-E-194-017.

The authors are with the Institute of Computer Science and Information Engineering, National Chung Cheng University, Taiwan 621, R.O.C. (e-mail: jjleou@cs.ccu.edu.tw).

Publisher Item Identifier S 1051-8215(98)02653-6.

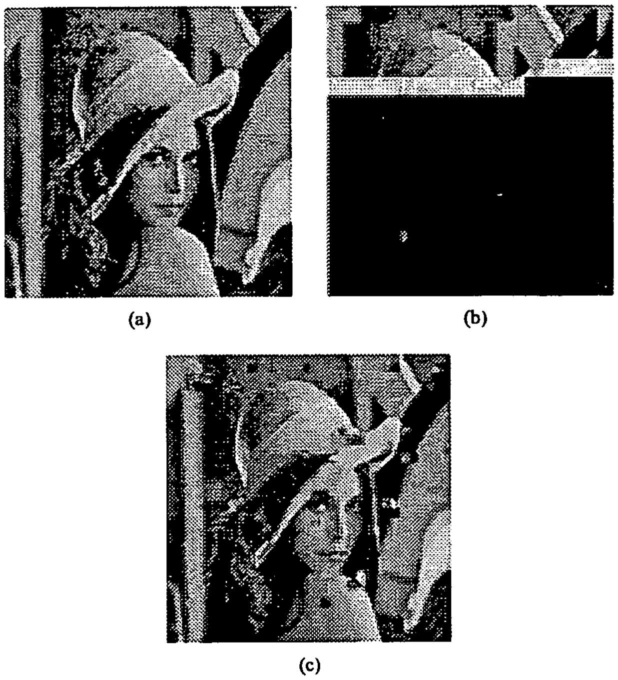


Fig. 1. The JPEG image "Lenna" with bit error rate (BER) = 0.1%: (a) the original image, (b) the corrupted image without the restart capability, and (c) the corrupted image with the restart capability.

uncorrupted blocks outside a corrupted block as the estimate for the missing dc coefficient and uses zeros for the ac coefficients; and 3) mean-prediction, which uses the mean of the external boundary samples of uncorrupted blocks outside a corrupted block as the estimate for the missing dc coefficient and uses two-dimensional polynomial predictions provided by the JPEG compression standard [6] to estimate the first five ac coefficients. Although various error concealment approaches can reduce the visual effect induced by transmission errors, the corresponding processed (reconstructed) images are usually blurred and transmission errors are not corrected at all.

For the detection and correction approach, based on the redundancy information inherent in neighboring pixels, the constraints on compressed image data, and the accompanied statistical property changes of compressed image data as well as the reconstructed image owing to transmission errors, transmission errors can be directly detected and corrected, without embedding redundant bits. The detection and correction approaches for several types of images have been proposed [19]–[26]. In this study, the detection and correction approach to transmission errors in JPEG images using the sequential DCT-based mode of operation is proposed.

In the proposed approach, the restart capability of JPEG is enabled, i.e., the eight unique restart markers RST_m, $m = 0, \cdots, 7$ are periodically inserted into the JPEG compressed image bitstream. The eight unique restart markers can be found by the proposed restart marker determination procedure. Transmission errors in a JPEG image are sequentially detected both when the JPEG image is under decoding and after the JPEG image has been decoded. When the JPEG image is under decoding, transmission errors are detected by checking a set of error detection conditions derived from the redundant information inherent within the image and the constraints imposed on the JPEG compressed image bitstream. After the JPEG image has been decoded, transmission errors are detected by the redundant information inherent within the image and the accompanied property changes of compressed image data as well as the reconstructed image owing to transmission errors. If a transmission error or equivalently a corrupted restart interval is detected, the proposed error correction approach simply performs a sequence of bit inversions and redecoding operations on the corrupted restart interval and selects the "best" feasible redecoding solution by applying a proposed cost function for error correction.

Here a transmission error may be a single bit error, which is mainly caused by system imperfections, or a burst error, which is caused by maintenance actions, such as protection switching [27]. To simplify the processing steps in the proposed approach, a burst error containing several error bit segments is treated as several transmission errors, i.e., a transmission error is equivalently either a single bit error or a burst error containing N successive error bits. For simplicity, four assumptions are made as follows: 1) the headers of a JPEG image are correctly received; 2) within a JPEG image, each restart interval followed by a restart marker contains at most a transmission error; 3) the first restart interval of a JPEG image is correctly received; and 4) if a restart marker is corrupted, the transmission error is always a single-bit error.

The paper is organized as follows. A brief overview of the JPEG compression standard is described in Section II. The proposed detection and correction approach to transmission errors in JPEG images is addressed in Section III. Simulation results are included in Section IV, followed by concluding remarks.

II. The JPEG Compression Standard

A. Overview of JPEG

The JPEG compression standard has four modes of operation [5], [6], namely, the sequential DCT-based, progressive DCT-based, sequential lossless, and hierarchical modes of operation. In this study the sequential DCT-based mode of operation (the most popular mode of operation) is treated. For the encoder of the sequential DCT-based mode of operation, 8×8 sample blocks are typically input block by block from left to right, and block-row by block-row from top to bottom. After a block has been transformed by the forward DCT, quantized and prepared for entropy encoding, all the 64 quantized DCT coefficients can be immediately entropy encoded and output as part of the compressed image data, thereby minimizing coefficient storage requirements. After quantization the difference between the current quantized dc coefficient and that of the previous block is encoded, and all the 63 quantized ac coefficients are converted into a "zig-zag" sequence. Each step of the decoder performs essentially the inverse of its corresponding main procedure within the encoder.

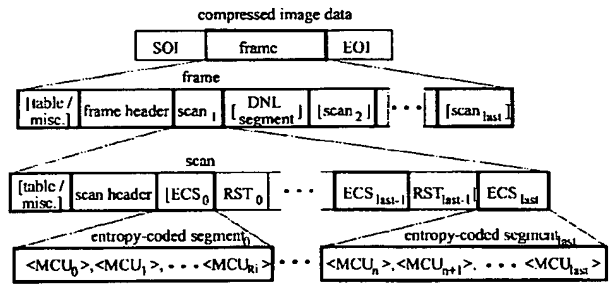


Fig. 2. The high-level syntax of JPEG for the sequential DCT-based, progressive DCT-based, and lossless modes of operation.

The JPEG compression standard specifies two entropy coding schemes—Huffman coding and arithmetic coding. Here Huffman coding as well as the default Huffman tables are employed.

B. Syntax and Data Organization of JPEG

For a DCT-based codec, the data unit is an 8×8 block of samples and data units are assembled into groups called minimum coded units (MCU's). If the source image contains only one component and the data are noninterleaved, the MCU is one data unit. For interleaved data, the MCU is the sequence of data units defined by the sampling factors of the component [5], [6].

Markers serve to identify the various structural parts of the compressed data formats. Most markers start marker segments containing a related group of parameters; some markers stand alone. A marker segment consists of a marker followed by a sequence of related parameters. The first parameter is the two-byte length parameter, which encodes the number of bytes in the marker segment. The marker segments identified by the start of frame (SOF) and start of scan (SOS) marker codes are referred to as the frame header and the scan header, respectively.

JPEG divides the compression sequence into frames and scans. For nonhierarchical mode of operation, the frame defines the basic attributes of the image, including size, number of components, precision, and entropy-coding technique. For nonhierarchical encoding, only a single frame is allowed. In addition to the frame and scan headers, a number of other marker segments are needed to define entropy-coding tables, quantization tables and parameters.

As shown in Fig. 2, the high-level syntax of JPEG specifies the order of the high-level constituent parts of the interchange format for all nonhierarchical encoding processes. In Fig. 2, three markers can be identified: 1) SOI (start of image) indicates the start of a compressed image; 2) EOI (end of image) indicates the end of a compressed image; and 3) RST_m (restart marker) is a marker placed between entropy-coded segments only if the restart capability is enabled. In JPEG, the eight unique restart markers (RST_m, $m = 0, 1, 2, \cdots, 7$) repeating in sequence from 0 to 7 provide a modulo 8 restart interval count. In this study, we consider the case that a frame shall contain one single scan with interleaved ordering, the restart capability is enabled, and a restart interval contain only one MCU including six data units (four Y blocks, one U block, and 1one V block). It is noted that the JPEG compression standard uses the YUV color space representation.

III. Proposed Detection and Correction Approach to Transmission Errors in JPEG Images

As shown in Fig. 3, the top three levels of the high-level syntax of JPEG can be translated into three decoding procedures, namely, Decode_image, Decode_frame, and Decode_scan, in the JPEG decoder. Based on the assumption that all the headers and the first restart interval of a JPEG image are correctly received, the proposed error detection and correction approach is performed in the Decode_scan procedure.

A. Proposed Error Detection Approach

1) Restart Marker Determination and Error Detection for a Decoding JPEG Image: In this study, transmission errors in a JPEG image are sequentially detected both when the JPEG image is under decoding and after the JPEG image has been decoded. When a JPEG image is under decoding, the restart marker may be corrupted and part of the compressed image bitstream within a corrupted restart interval may generate a "false" restart marker or a "near" restart marker, where a "near" restart marker is part of the compressed image bitstream which has only one bit difference (the Hamming distance = 1) from a correct restart marker. Therefore, it is necessary to find the "correct" restart markers within the compressed image bitstream. In this study, either a restart marker or a "near" restart marker is treated as a candidate of a restart marker. To find a restart marker $\mathrm{RST_i}$, starting from the following bit of the previously-determined restart marker $\mathrm{RST_{i-1}}$, all the candidates of the restart marker $\mathrm{RST_i}$ and a series of other restart markers scanned in the compressed image bitstream are recorded. Then starting from the following bit of $\mathrm{RST_{i-1}}$, we begin to decode the compressed image bitstream until a complete MCU is decoded. If a candidate of the restart marker $\mathrm{RST_i}$ appears at the position just after the completely-decoded MCU, the candidate will be treated as the restart marker $\mathrm{RST_i}$. If no candidate appears at the position just after the completely decoded MCU, starting from the following bit of $\mathrm{RST_{i-1}}$, we count the number ($\mathrm{N_c}$) of candidates of the restart marker $\mathrm{RST_i}$ and the number ($\mathrm{N_o}$) of other restart markers until an exact $\mathrm{RST_i}$ is found in the compressed image bitstream. If $\mathrm{N_c}+\mathrm{N_o} \leq 8$, the exact $\mathrm{RST_i}$ immediately-found is determined to be the restart marker $\mathrm{RST_i}$ and the "current" restart interval between $\mathrm{RST_{i-1}}$ and $\mathrm{RST_i}$ is determined to be a corrupted one. If $\mathrm{N_c} + \mathrm{N_o} > 8$, the first candidate is determined to be the restart marker $\mathrm{RST_i}$ and the "current" restart interval is also determined to be a corrupted one. Additionally, because the minimum number of bits of a completely decoded MCU is equal to 62, we should "check" if the total number ($\mathrm{N_b}$) of bits of the compressed image bitstream between the previously-determined restart marker $\mathrm{RST_{i-1}}$ and any candidate or any restart marker is smaller than 62. The candidates or restart markers with $\mathrm{N_b} < 62$ will be ignored in the proposed procedure.

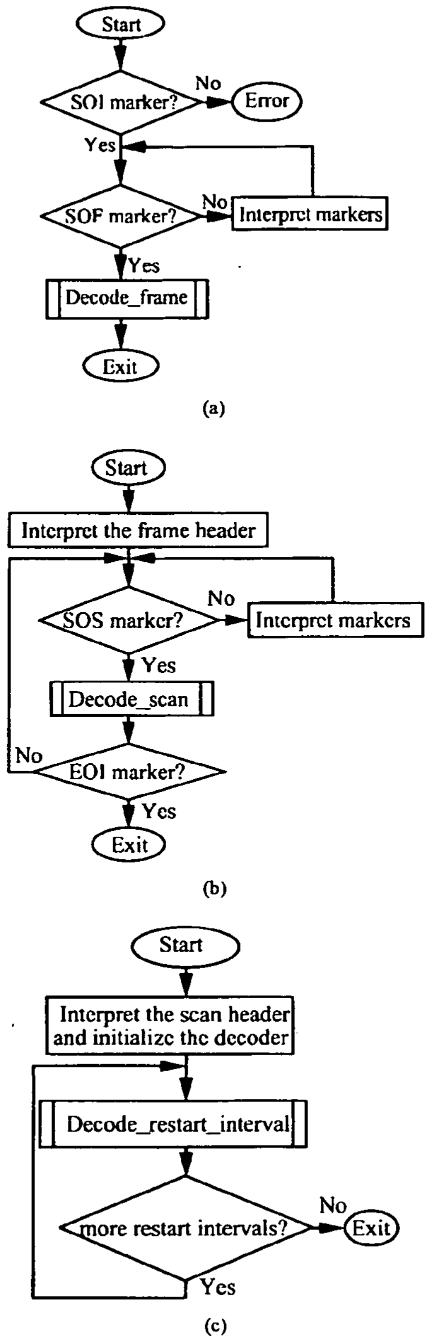


Fig. 3. The three decoding procedures of the decoder for the top three levels of the syntax of JPEG.

When a JPEG image is under decoding, based on the redundant information inherent within the image and the constraints imposed on the JPEG compressed image bitstream, the set of error detection conditions for detecting transmission errors can be listed as follows.

1) The decoder fails to detect a restart marker at the expected position requiring resynchronization within the compressed image bitstream.
2) Physically impossible image data are decoded. For example, a magnitude beyond the range of allowed values is detected.
3) The number of decoded coefficients within a block is larger than 64.
4) The number of received bits within an MCU is larger than 9448.

Once any of the error detection conditions is satisfied, the current restart interval is determined to be a corrupted one.

2) Error Detection for a Decoded JPEG Image: After a JPEG image has been decoded, we will detect transmission errors or corrupted blocks (corrupted restart intervals) again. To detect corrupted blocks (corrupted restart intervals) the AID (average intersample difference) of a block ($N \times M$ in size) using a local coordinate system is given by [29] as shown in (1) at the bottom of the page where $I(x, y)$ is the pixel value of position (x, y). The corrupted blocks can be classified into two categories, namely, smooth corrupted blocks and unsmooth corrupted blocks. The AID value of a smooth corrupted block will be relatively small, whereas the AID value of an unsmooth block, containing grids, strips, and edges, will be relatively large. Additionally, there always exist clear edges along the boundaries between a corrupted smooth block and its neighboring uncorrupted blocks. Here two measures used to detect possible corrupted blocks include: 1) the number of edge points along block boundaries and 2) the AID value of a block. Here edge points along block boundaries are detected by the Sobel operator followed by binary thresholding with a threshold T [28]. If the number of edge points on the block boundaries of a block is greater than a threshold T_e or the AID value of the block is larger than a threshold T_a, the block is determined to be a corrupted block. In this study, because a restart interval contains an MCU encoding sequentially six entropy-coded blocks (four Y blocks, one U block, and one V block), a restart interval is determined to be a corrupted one if one or more blocks among the six entropy-coded blocks is corrupted. So a restart interval will be marked as a corrupted one if it contains one or more corrupted blocks. The proposed error detection procedure for a decoded JPEG image will be performed at most three times on a decoded JPEG image.

B. Proposed Error Correction Approach

After the proposed restart marker determination and error detection procedure has been performed on the compressed

$$\mathrm{AID} = \frac{\sum_{y=0}^{N-1}\sum_{x=1}^{M-1}|I(x, y) - I(x-1, y)| + \sum_{x=0}^{M-1}\sum_{y=1}^{N-1}|I(x, y) - I(x, y-1)|}{N \times (M-1) + M \times (N-1)} \tag{1}$$

image bitstream under decoding, both the corrupted blocks and the corrupted restart intervals within the compressed image bitstream are thus determined. To correct transmission errors, a sequence of bit inversions is performed on each corrupted restart interval to generate all feasible redecoding solutions and then a proposed cost function for error correction is used to select the "best" redecoding solution among all feasible redecoding solutions. Here the proposed cost function for error correction may combine the number of edge points on the block boundaries of the block and/or the AID value of the block. Fig. 4(a) shows a block C and its four-connected neighboring blocks A, B, D, and E. To evaluate the number of edge points between neighboring blocks, a measure called the average intersample difference across block boundary (boundaries) AID_b can be defined as the total absolute intersample differences between the relating points of neighboring blocks divided by the total number of boundary points. As the illustrative example shown in Fig. 4(b), AID_b for the two block boundaries, b(C,A) and b(C,B), of block C is given by

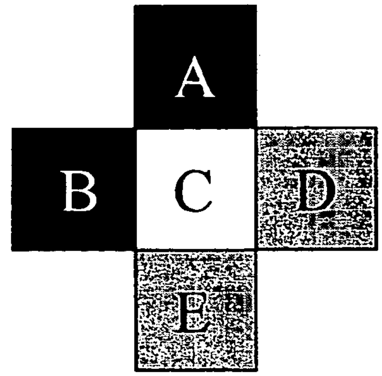
$$\mathrm{AID}_b(C\colon A, B) = \frac{1}{2M}\left[\sum_{i=1}^{M}|c_i - b_i| + \sum_{i=1}^{M}|c_i^* - a_i^*|\right]. \quad (2)$$

AID_b (C: A, B, D, E) for the four block boundaries, b(C,A), b(C,B), b(C,D), and b(C,E), of block C can be similarly defined. Using the average intersample difference across block boundaries $\mathrm{AID}_b(C{:}*)$ of block C and the AID value of block C, the proposed cost function $\mathrm{CF}(C)$ for block C is given by

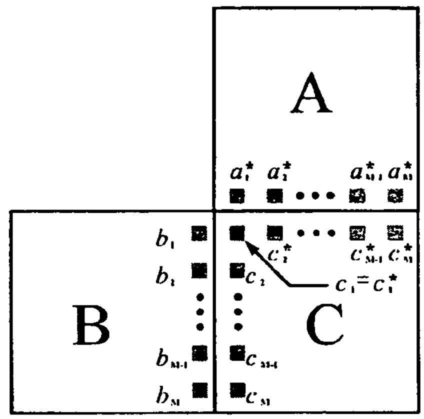
$$\mathrm{CF}(C) = w_1 \times [\mathrm{AID}_b(C:*)]^2 + w_2 \times [\mathrm{AID}(C)]^2 \quad (3)$$

where w_1 and w_2 are two weighting coefficients. Here w_1 and w_2 can be adaptively adjusted by the total number of encoded bits within an MCU. When the total number of encoded bits of an MCU is a small value, i.e., the MCU contains smooth blocks with small AID values, then w_2 is set to a larger value, and vice versa. For an MCU containing six blocks (four Y blocks, one U block, and one V block), the proposed cost function of a restart interval (here an MCU) is set to the sum of the cost function values of the six blocks. For a corrupted restart interval, if two or more feasible redecoding solutions are generated, the feasible redecoding solution having the smallest proposed cost function value will be determined to be the "best" redecoding solution of the restart interval (or MCU). Here a "feasible" redecoding solution means that the redecoding solution exactly reconstructs six blocks of an MCU and none of the four error detection conditions for a decoding JPEG image is satisfied.

When a JPEG image is under decoding, the previously decoded restart intervals are used as the reference to detect and correct the current corrupted restart interval. After the JPEG has been decoded both the preceding and following restart intervals of the current restart interval can be used as the references to detect and correct the current restart interval. In this study, if the total number of corrupted restart intervals within a JPEG image is larger than a threshold (here ten), the error detection and correction procedure for a decoded JPEG image will be iteratively performed on the "previously-corrected" JPEG image until either the total number of "corrupted" restart intervals is smaller than a threshold (here ten) or the number of iterations is larger than three. In the meantime, as shown in Fig. 4, instead of only the neighboring blocks A and B employed in the proposed error detection and correction procedure for a decoding JPEG image, the neighboring blocks of block C employed in the proposed error detection and correction procedure for a decoded JPEG image include all the four-connected neighboring blocks A, B, D, and E. That is, AID_b $(C\colon A, B)$ and AID_b $(C\colon A, B, D, E)$ will be employed within the two procedures, respectively. As a summary, the proposed error detection and correction approach to transmission errors in a JPEG image is summarized in Fig. 5.



(a)



(b)

Fig. 4. (a) The four-connected neighboring blocks A, B, D, and E of block C and (b) the relating points for computing AID_b(C: A, B).

Referring to Fig. 5, the proposed error detection and correction procedure for a decoding JPEG image will be performed on the compressed image bitstream one time only. The proposed error detection and correction procedure for a detected JPEG image will be performed on the decoded JPEG image at most three times. Both the two error detection and correction

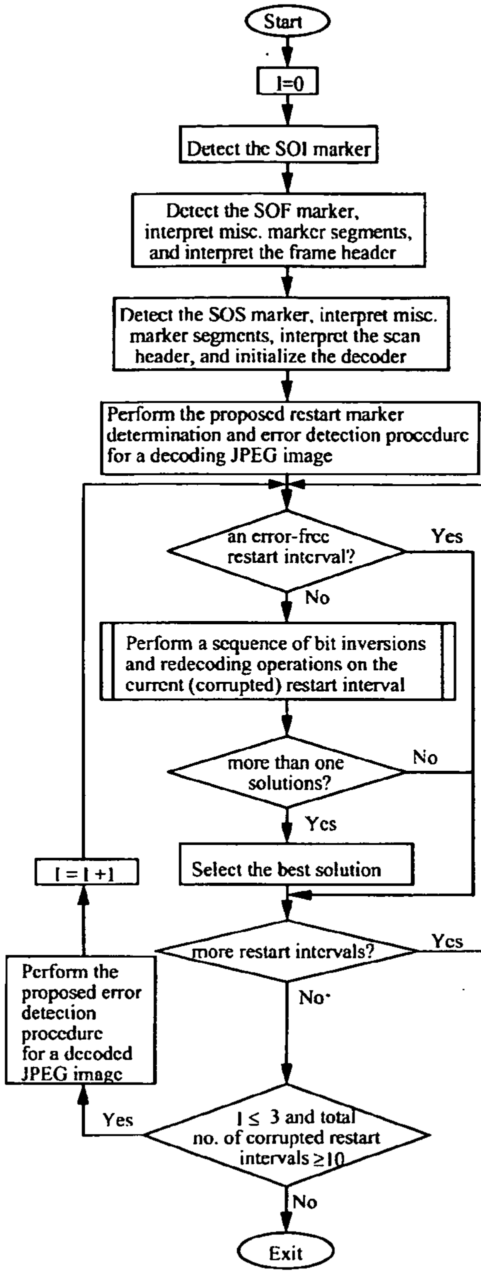


Fig. 5. The proposed error detection and correction approach to transmission errors in a JPEG image.

procedures will be performed on each restart interval of a corrupted JPEG image.

C. Computational Complexity Analysis

To analyze the computational complexity of the proposed approach, several notations are defined as follows. Let the total number of restart intervals (or MCU's) in a JPEG image, the total number of corrupted restart intervals detected by the proposed error detection procedure for a decoding JPEG image, the total numbers of corrupted restart intervals detected by the proposed error detection procedure for a decoded JPEG image in the three iterations be denoted as N_{rs}, N_{ca}, N_{cb1}, N_{cb2}, and N_{cb3}, respectively. Let the time used to match a restart marker in the compressed image bitstream, the time used to decode/redecode a restart interval in the compressed image bitstream, the time used to compute the number of edge points along block boundaries, the time used to compute the AID value of a block, the time used to compute the AID_b value of a block be denoted as t_{mat}, t_{dec}, t_{edge}, t_{AID}, and t_{AID_b}, respectively. Let the total number of bits within a restart interval be denoted as N_b. In the worst case, the time used to perform the proposed restart marker determination and error detection procedure for a decoding JPEG image is approximately equal to $N_{rs}[(8N_c + N_o)t_{mat} + t_{dec}]$, where the number 8 denotes the eight candidates of each restart marker. The time used to perform the proposed error correction approach on the corrupted restart intervals within a decoding JPEG image is approximately equal to $N_{ca}[N_b + (N_b - 1) + (N_b - 2)](t_{dec} + t_{AID} + t_{AID_b})$ where N_b, $N_b - 1$, and $N_b - 2$ denote the total numbers of possible transmission errors within a restart interval containing single-error bit, two successive error bits, and three successive error bits, respectively. The time used to perform the proposed error detection procedure for a decoded JPEG image is approximately equal to $N_{rs}(t_{edge} + t_{AID})$. The times used to perform the proposed error correction approach on the corrupted restart intervals within a decoded JPEG image are approximately equal to $N_{cbi}[N_b+(N_b-1)+(N_b-2)](t_{dec}+t_{AID} + t_{AID_b}), i = 1, 2, 3$, respectively. Because the proposed error detection and correction procedure for a decoded JPEG image will be performed at most three times, for the worst case the total processing time used to process a JPEG Image by the proposed approach is approximately equal to $N_{rs}[(8N_c+N_o)t_{mat} + t_{dec}] + 3N_{ca}N_b(t_{dec} + t_{AID} + t_{AID_b}) + 3N_{rs}(t_{edge} + t_{AID}) + 3N_b(N_{cb1}+N_{cb2}+N_{cb3})(t_{dec}+t_{AID}+t_{AID_b})$ where the former two terms denote the time for performing the proposed error detection and correction procedure for a decoding JPEG image, and the latter two terms denote the times for performing the proposed error detection and correction procedure for a decoded JPEG image (three times). It is noted that the numbers of corrupted restart intervals detected by the two proposed error detection procedures, i.e., N_{ca}, N_{cb1}, N_{cb2}, and N_{cb3}, are random variables. The two numbers, N_c and N_o (defined in Section III-A1), for restart marker determination are random variables. The time used to decode a restart interval, t_{dec}, and the total number of bits within a restart interval, N_b, are also random variables. It is noted that the total processing time used to process a JPEG image is applicable for the case that each transmission (burst) error contains at most three successive error bits. The total processing times used to process a JPEG image for other cases can be similarity analyzed. Several processing times, such as the time used to process the headers in JPEG, are ignored in the computational complexity analysis.

IV. Simulation Results

The proposed error detection and correction approach to transmission errors in JPEG images has been implemented on a SUN SPARC-5 workstation using the C programming language. Five test images "Lenna," "Baboon," "Peppers,"

"Airplane," and "Sailboat" with different BER (bit error rates) and different N_{ave} (average length of burst errors) are used to evaluate the performance of the proposed approach. The PSNR (peak signal to noise ratio) is employed in this study as the objective performance measure for three individual components (Y, U, V) in each JPEG image. The average PSNR, denoted by $PSNR_a$, for a JPEG image is given by

$$PSNR_a = (4 \times PSNR_Y + PSNR_U + PSNR_V)/6 \quad (4)$$

where $PSNR_Y$, $PSNR_U$, and $PSNR_V$ are the PSNR values for the Y, U, and V components of a JPEG image, respectively. The average length of burst errors, N_{avc}, is given by

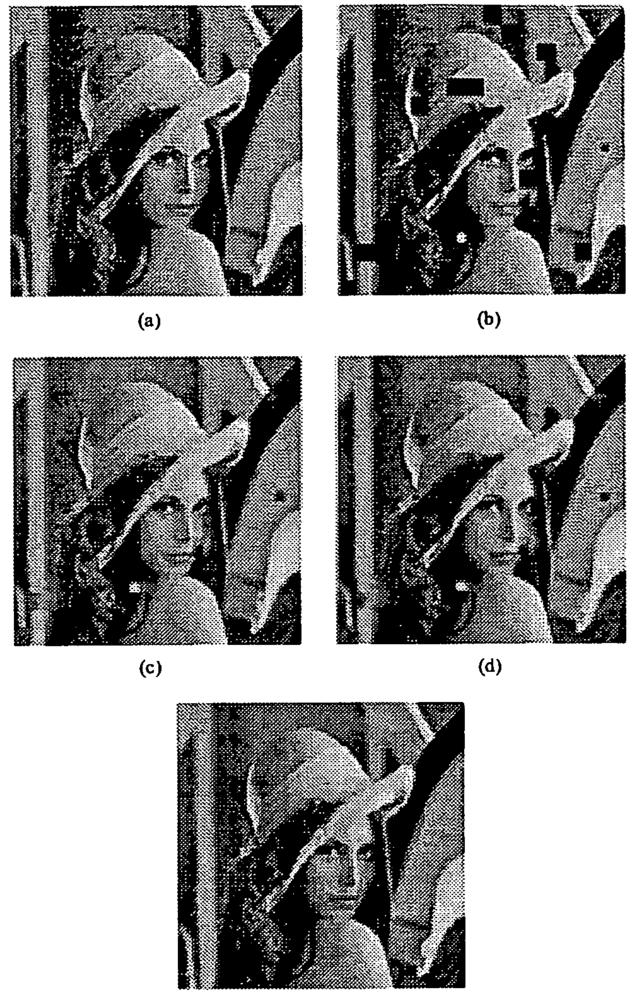
$$N_{avc} = \sum_{i=1}^{N} i \times P_i \quad (5)$$

where P_i is the probability of a burst error containing i successive error bits with $\sum_{i=1}^{N} P_i = 1$.

In this study, N (the maximum number of successive error bits of a burst error) is set to three and N_{avc} will lie within the range (1, 3). To test the five test images with $N_{ave} = 1$, 2, and 3, the corresponding three sets of probabilities P_i are set to {1, 0, 0}, {0.3, 0.4, 0.3}, and {0, 0, 1}, respectively. The threshold T used to detect edge points is set to 60 for the Y component and identically 40 for the U and V components. The threshold T_e used to confirm edges along block boundaries is set to ten. The threshold T_a for the AID value of a block used to detect unsmooth blocks is set to 45 for the Y component and identically 25 for the U and V components. The two weighting coefficients, w_1 and w_2, of the proposed cost function for error correction are set to {0.65, 0.35} and {0.5, 0.5} for the "long" and "short" restart intervals, respectively. The above thresholds and weighting coefficients are determined by means of experiments.

In terms of $PSNR_a$ in decibels, the comparative simulation results for the three test images, "Lenna," "Baboon," and "Peppers," with different BER and N_{ave} of three error concealment approaches, namely, zero-substitution, mean-substitution, and mean-prediction, and the proposed approach are illustrated in Tables I–III, respectively. The comparative simulation results for the other two test images, "Airplane," and "Sailboat," can be found in Han [29], and thus omitted here. In Tables I–III, each "*" indicates the case that on the average a restart interval followed by a restart marker contains two or more transmission errors and the "no loss" column denotes the corresponding PSNR values for the error-free reconstructed images. As subjective measure of the quality of the processed (reconstructed) images, the original and processed images (the Y component) by the three error concealment approaches and the proposed approach with BER = 0.1% and $N_{ave} = 2$ for the three test images "Lenna," "Baboon," and "Peppers," are shown in Figs. 6–8, respectively.

The total processing time for a corrupted JPEG image by the proposed approach depends on the content of the JPEG image, BER, N_{ave} (average length of burst errors), and the distribution of transmission errors within the corrupted image.



(a) (b) (c) (d) (e)

Fig. 6. The JPEG image (the Y component) "Lenna" with BER = 0.1% and $N_{avc} = 2$: (a) the original image, (b)–(e) the processed images by the zero-substitution approach, the mean-substitution approach, the mean-prediction approach, and the proposed approach, respectively.

A smooth image will take less time, whereas an unsmooth image will take more time. In practice, when a SUN SPARC-5 workstation is employed, the total processing time for the corrupted image "Lenna" with BER = 0.01% and $N_{ave} = 1$ is less than three min. The total processing time for the corrupted image "Baboon" with BER = 0.4% and $N_{ave} = 1$ (an asterisk case) is around 30 min.

Based on the simulation results obtained in this study, we have the following observations.

1) When BER is a small value, a corrupted JPEG image will contain a "small" number of (sparse) corrupted restart intervals. In this situation, the performances (PSNR) of the mean-substitution as well as mean-prediction approaches are comparable or slightly inferior to that of the proposed approach. But when BER is a large value, a corrupted JPEG image will contain a "large" number of (dense and neighboring) corrupted restart intervals. In this situation, the performances (PSNR) of the mean-substitution as well as

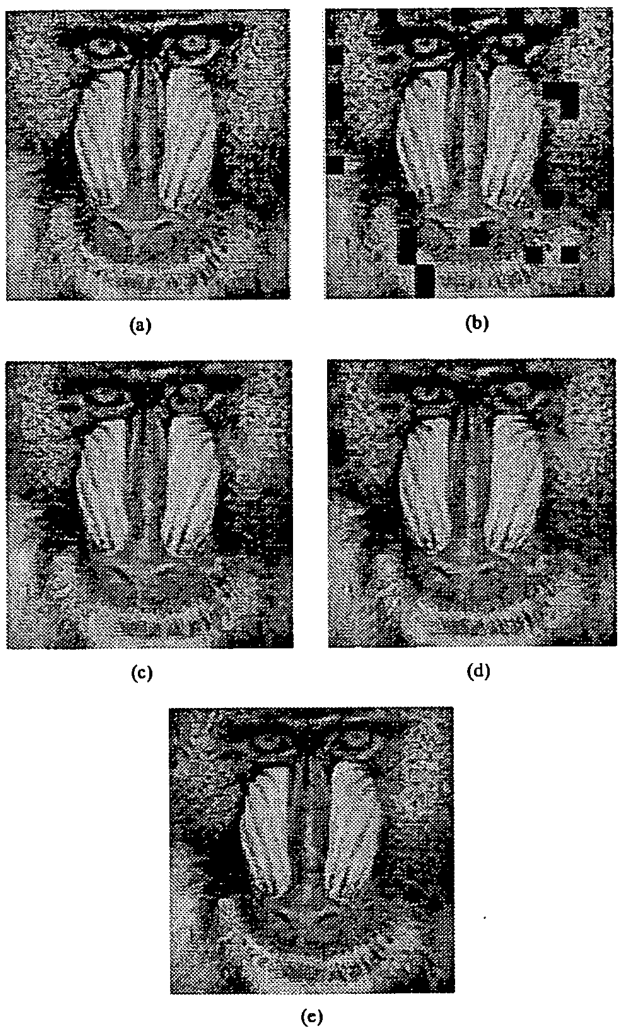


Fig. 7. The JPEG image (the Y component) "Baboon" with BER = 0.1% and $N_{ave} = 2$: (a) the original image, (b)–(e) the processed images by the zero-substitution approach, the mean-substitution approach, the mean-prediction approach, and the proposed approach, respectively.

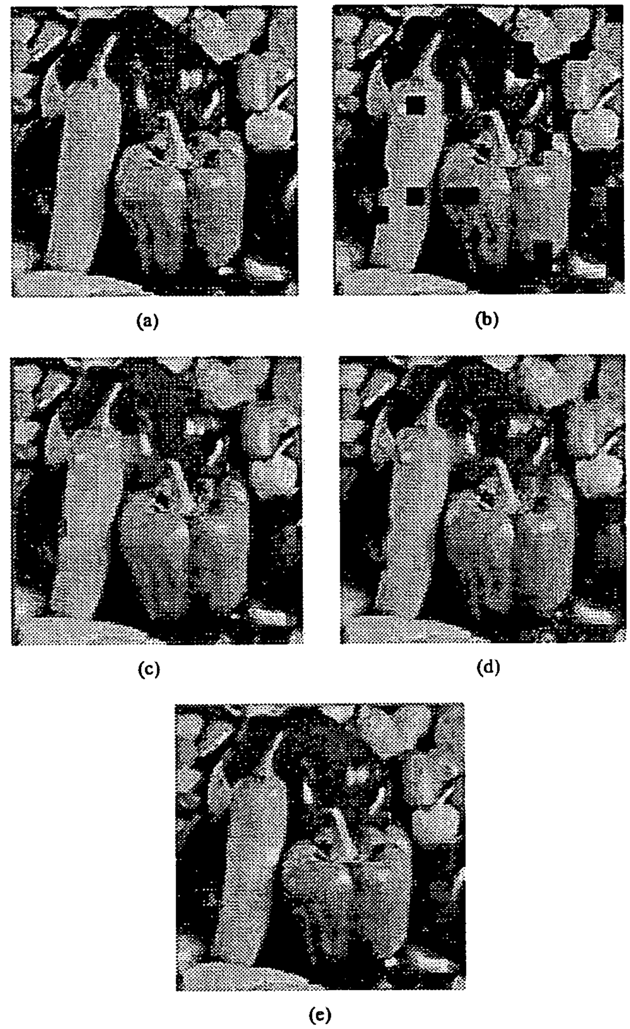


Fig. 8. The JPEG image (the Y component) "Peppers" with BER = 0.1% and $N_{ave} = 2$: (a) the original image, (b)–(e) the processed images by the zero-substitution approach, the mean-substitution approach, the mean-prediction approach, and the proposed approach, respectively.

mean-prediction approaches are relatively inferior to that of the proposed approach.

2) The thresholds, T, T_e, and T_a, and the two weighting coefficients, w_1 and w_2, used in this study will affect the processed results by the proposed approach. That is because whether a block is a corrupted block or not is affected by T, T_e, and T_a. The "best" feasible redecoding solution is determined by the proposed cost function for error correction, which is affected by w_1 and w_2. In general, the thresholds T for detecting edge points, T_e for confirming edges, T_a for detecting unsmooth blocks (via AID) may be set to relatively large values for the relatively unsmooth Y image (component) and relatively small values for the relatively smooth U or V image (component). But T, T_e, and T_a may be invariant for different test images (different restart intervals). On the other hand, when the total number of encoded bits of MCU is a small value (a smooth restart interval), w_2 is set to a larger value ($w_1 + w_2 = 1$), and vice versa. That is, w_1 and w_2 may be adaptively adjusted for different restart intervals. In this study, only two different sets of $\{w_1, w_2\}$ are employed. The thresholds and weighting coefficients are determined by experiments. The acceptable ranges of the thresholds and weighting coefficients are ±5% deviations from the values used in this study.

3) If the compressed image bitstream contain many candidates for each restart marker, the possibility of loss synchronization will increase.

4) Because in this study a transmission error is equivalently either a single-bit error or a burst error containing N successive error bits, under the same BER, the larger N_{ave} will result in a smaller number of transmission errors. Under the same BER and using the same approach, a corrupted JPEG image with the larger N_{ave} will produce the better processed image with a higher $PSNR_a$.

TABLE I
THE SIMULATION RESULTS $PSNR_a$ FOR THE IMAGE "LENNA" WITH DIFFERENT BER AND N_{ave} OF THE THREE ERROR CONCEALMENT APPROACHES AND THE PROPOSED APPROACH

BER	N_{ave}	$PSNR_a$ (dB)				
		No loss	Zero-Subs.	Mean-Subs.	Mean-Pred.	Proposed
0.01%	1	32.3	23.8	29.2	29.4	30.8
	2	32.3	24.5	30.7	30.7	30.8
	3	32.3	27.1	30.8	30.8	30.8
0.03%	1	32.3	19.8	27.1	27.1	30.6
	2	32.3	21.0	30.2	30.3	30.7
	3	32.3	23.0	29.8	29.9	30.8
0.05%	1	32.3	19.5	28.7	28.7	30.6
	2	32.3	19.3	29.4	29.4	30.8
	3	32.3	21.1	29.8	29.8	30.8
0.07%	1	32.3	18.7	27.5	27.5	30.7
	2	32.3	16.9	27.0	27.1	30.8
	3	32.3	19.4	29.8	29.9	30.6
0.09%	1	32.3	15.8	26.6	26.7	30.4
	2	32.3	18.0	28.0	28.0	30.7
	3	32.3	20.2	27.5	27.6	30.7
0.1%	1	32.3	15.1	26.4	26.5	30.5
	2	32.3	17.0	26.9	26.9	30.7
	3	32.3	16.9	27.1	27.2	30.8
0.2%	1	32.3	12.6	23.5	23.6	30.6
	2	32.3	14.7	24.1	24.2	30.6
	3	32.3	16.2	27.6	27.6	30.0
0.3%	1	32.3	10.1	20.4	20.4	29.8
	2	32.3	12.6	24.7	24.8	30.3
	3	32.3	14.1	24.3	24.3	30.5
0.4%	1	32.3*	9.7*	19.9*	19.8*	29.3*
	2	32.3	11.2	22.2	22.3	30.2
	3	32.3	12.5	22.9	22.9	30.1

TABLE II
THE SIMULATION RESULTS $PSNR_a$ FOR THE IMAGE "BABOON" WITH DIFFERENT BER AND N_{ave} OF THE THREE ERROR CONCEALMENT APPROACHES AND THE PROPOSED APPROACH

BER	N_{ave}	$PSNR_a$ (dB)				
		No loss	Zero-Subs.	Mean-Subs.	Mean-Pred.	Proposed
0.01%	1	27.5	20.1	26.1	26.1	25.2
	2	27.5	20.0	26.8	26.8	25.3
	3	27.5	21.7	26.9	26.9	25.3
0.03%	1	27.5	17.6	25.4	25.5	25.3
	2	27.5	18.7	26.3	26.3	25.3
	3	27.5	19.4	26.0	26.0	25.3
0.05%	1	27.5	14.7	24.2	24.2	25.3
	2	27.5	17.4	25.8	25.8	25.3
	3	27.5	17.3	26.0	26.1	25.3
0.07%	1	27.5	13.7	24.0	24.0	25.2
	2	27.5	15.8	24.6	24.7	25.0
	3	27.5	16.6	25.4	25.5	25.3
0.09%	1	27.5	12.7	23.3	23.3	25.2
	2	27.5	15.3	24.7	24.7	25.3
	3	27.5	15.0	24.8	24.8	25.3
0.1%	1	27.5	12.8	23.5	23.5	25.4
	2	27.5	15.1	24.6	24.6	25.3
	3	27.5	15.9	25.0	25.0	25.2
0.2%	1	27.5*	10.1*	21.4*	21.9*	25.0*
	2	27.5	11.7	22.3	22.4	25.1
	3	27.5	13.0	22.8	22.9	25.3
0.3%	1	27.5*	9.3*	20.5*	21.2*	24.8*
	2	27.5	10.4	21.4	21.4	25.1
	3	27.5	11.3	21.6	21.6	25.0
0.4%	1	27.5*	9.0*	20.0*	20.5*	24.5*
	2	27.5*	10.0*	21.3*	21.3*	24.9*
	3	27.5	11.0	21.6	21.6	25.1

5) If the original image is relatively unsmooth, the total processing time of the proposed approach will increase. That is because, for an unsmooth image, the larger number of irregular boundaries will actually result in the larger number of times of applying the proposed error detection and correction procedure for a decoded JPEG image.
6) If all the assumptions are valid, for a particular image, the $PSNR_a$ values of the processed images with different BER and N_{ave} by the proposed approach are approximately constant, as the simulation results shown in Tables I, II, and III. That is, the proposed approach is very robust. However, for the three error concealment approaches for comparison, when BER is increased, the corresponding $PSNR_a$ values will decrease accordingly.

When an original image is relatively unsmooth, i.e., the texture of each block of the image is irregular and unsimilar to those of its neighboring blocks, the proposed error detection procedure for a decoded JPEG image may make a mistake and the performance of the proposed approach will be relatively degraded accordingly. As an illustrative example shown in Fig. 8, there is a square on the right of the long pepper that is "incorrectly detected" by the proposed error detection procedure for a decoded JPEG image and "incorrectly corrected" by the proposed error correction approach. In this situation, the $PSNR_a$ value of the processed image by the proposed approach is approximately equal to that by the mean-prediction approach (even that by the mean-substitution approach). Because the fine details of the processed images by the three error concealment approaches for comparison are almost lost, the quality of the processed image by the proposed approach is still better than that by the three error concealment approaches for comparison. If the JPEG images are very noisy, i.e., most restart intervals contain two or more transmission errors, such as the asterisk cases in Tables I–III, the proposed error detection approaches for both a decoding JPEG image and a decoded JPEG image are still very effective in detecting transmission errors and the proposed error correction approach can not correct completely all the transmission errors. However, using the proposed cost function for error correction, the proposed error correction approach will find the "best" feasible redecoding solution, which is still superior to that of the three error concealment approaches for comparison. As a whole, the proposed approach is superior to the three error concealment approaches for comparison.

Under the condition that one restart interval contains only one MCU, the total numbers of encoded bits and average bit rate increase for the five test images with/without the restart capability are listed in Table IV. That is, the average bit rate increase for the five test images are less than 9.326%, which are modest and acceptable for practical applications. In this study, N (the maximum number of successive error bits of

TABLE III
THE SIMULATON RESULTS $PSNR_a$ FOR THE IMAGE "PEPPERS" WITH DIFFERENT BER AND N_{ave} OF THE THREE ERROR CONCEALMNET APPROACHES AND THE PROPOSED APPROACH

BER	N_{ave}	$PSNR_a$ (dB)				
		No loss	Zero-Subs.	Mean-Subs.	Mean-Pred.	Proposed
0.01%	1	30.6	21.9	27.0	27.3	28.4
	2	30.6	22.0	27.6	27.7	28.4
	3	30.6	24.7	27.8	27.8	28.3
0.03%	1	30.6	19.1	26.0	26.0	28.3
	2	30.6	20.8	27.4	27.4	28.4
	3	30.6	20.4	27.2	27.3	28.2
0.05%	1	30.6	17.2	24.7	24.7	28.3
	2	30.6	19.6	26.7	26.9	28.4
	3	30.6	20.8	25.6	25.7	28.4
0.07%	1	30.6	16.9	24.9	25.0	28.0
	2	30.6	18.0	25.8	25.9	28.2
	3	30.6	19.2	26.7	26.9	28.2
0.09%	1	30.6	15.5	23.1	23.2	27.7
	2	30.6	16.7	25.5	25.5	28.1
	3	30.6	17.1	25.3	25.5	27.7
0.1%	1	30.6	15.6	23.5	23.6	27.8
	2	30.6	16.3	25.0	25.2	28.0
	3	30.6	20.3	26.6	26.6	28.3
0.2%	1	30.6	13.1	22.4	22.5	27.6
	2	30.6	14.0	23.7	23.8	27.8
	3	30.6	15.7	23.9	24.0	28.0
0.3%	1	30.6*	11.0*	20.8*	21.0*	27.2*
	2	30.6	12.9	22.0	22.0	27.6
	3	30.6	13.0	22.9	23.0	27.9
0.4%	1	30.6*	10.3*	19.8*	20.1*	27.1*
	2	30.6	11.3	20.9	21.0	27.8
	3	30.6	12.9	22.2	22.3	28.0

TABLE IV
THE TOTAL NUMBERS OF ENCODED BITS AND AVERAGE BIT RATE INCREASE (%) FOR THE FIVE TEST IMAGES WITH/WITHOUT THE RESTART CAPABILITY

Images	total no. of encoded bits		average bit rate increase
	without the restart capability	with the restart capability	
Lenna	74712	81680	9.326%
Airplane	75416	81280	7.776%
Baboon	127288	132880	4.393%
Sailboat	101440	107744	6.215%
Peppers	86648	93128	7.479%

a burst error) is set to three. Because each burst error may contain several error bit segments, each burst error may contain more than three error bits actually. This assumption is adequate for BER $< 0.2\%$. If BER $\geq 0.2\%$, the assumptions made in this study will be invalid, because in this situation the asterisk case will occur. Although N can be set to a larger number without much difficulty, both the computational complexity of the proposed approach and the total processing time used to process a corrupted JPEG image will moderately increase if N is increased.

In this study, it is assumed that the headers of a JPEG are correctly received. However, each bit of a compressed image bitstream may be corrupted by noise. In practice, if BER is very small (for example BER $< 0.01\%$), protection of header information is not greatly desirable. If BER is relatively large (for example BER $> 0.1\%$), protection of header information is necessary. However, protection of header information will lose the full compliance of the JPEG compression standard and increase the transmission bit rate.

V. CONCLUDING REMARKS

In this study, the detection and correction approach to transmission errors in JPEG images using the sequential DCT-based mode of operation is proposed. To cope with the synchronization problem in entropy-coded JPEG images, the restart capability of JPEG images is enabled, i.e., the eight unique restart markers RST_m, $m = 0, \cdots, 7$, are inserted periodically into JPEG images so that the synchronization problem can be solved. Here a transmission error is equivalently either a single-bit error or a burst error containing N successive error bits.

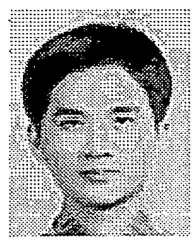
In the proposed approach, transmission errors in a JPEG image are sequentially detected both when the JPEG image is under decoding and after the JPEG image has been decoded. When a transmission error (or equivalently a corrupted restart interval) is detected, the proposed error correction approach simply performs a sequence of bit inversions and redecoding operations on the corrupted restart interval and selects the "best" feasible redecoding solution by applying a proposed cost function for error correction.

Based on the simulation results obtained in this study, the proposed approach can recover high-quality JPEG images from their corresponding corrupted JPEG images at BER's up to approximately 0.4%. Even for the asterisk cases in Tables I–III, where the processed images by the three error concealment approaches for comparison are relatively poor, the processed images by the processed approach are still relatively good with relatively high $PSNR_a$. This shows the feasibility of the proposed approach.

REFERENCES

[1] C. T. Chen and T. R. Hsing, "Review: Digital coding techniques for visual communications," *J. Visual Commun. Image Representation*, vol. 2, no. 1, pp. 1–16, 1991.
[2] M. Rabbani and P. W. Jones, *Digital Image Compression Techniques*. Bellingham, WA: SPIE Optical Engineering, 1991.
[3] D. Bodson, K. R. McConnell, and R. Schaphorst, *FAX: Digital Facsimile Technology and Applications*, 2nd ed. Boston, MA: Artech House, 1992.
[4] H. Hampel *et al.*, "Technical features of the JBIG standard for progressive bi-level image compression," *Signal Processing: Image Commun.*, vol. 4, no. 2, pp. 103–111, 1992.
[5] W. B. Pennebaker and J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. New York: Van Nostrand Reinhold, 1993.
[6] CCITT Recommendation T.81, "Digital compression and coding of continuous-tone still images," 1992.
[7] CCITT Recommendation H.261, "Video codec for audiovisual services at px64 kbit/s," 1990.
[8] ISO/IEC JTC1/SC29 WG11/602, Recommendation H.262, 1993.
[9] S. Lin and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*. Englewood Cliffs, NJ: Prentice-Hall, 1983.
[10] Y. Wang, Q. F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. Commun.*, vol. 41, pp. 1544–1551, Oct. 1993.
[11] H. Sun, K. Challapal, and J. Zdepski, "Error concealment in digital simulcast AD-HDTV decoder," *IEEE Trans. Consumer Electron.*, vol. 38, pp. 108–118, 1992.

[12] X. Lee, Y. Q. Zhang, and A. Leon-Garcia, "Information loss recovery for block-based image coding techniques—A fuzzy logic approach," *SPIE*, vol. 2094, pp. 529–540, 1993.
[13] K. H. Jung, J. H. Chang, and C. W. Lee, "Error concealment technique using projection data for block-based image coding," *SPIE*, vol. 2308, pp. 1466–1476, 1994.
[14] M. Ghanbari and V. Seferidis, "Cell-loss concealment in ATM networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 238–247, Mar. 1993.
[15] M. Wada, "Selective recovery of video packet loss using error concealment," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 807–814, June 1989.
[16] Q. F. Zhu, Y. Wang, and L. Shaw, "Coding and cell-loss recovery in DCT-based packet video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 3, pp. 248–258, Mar. 1993.
[17] S. H. Lee, J. S. Youn, S. H. Jang, and S. H. Jang, "Transmission error detection, resynchronization, and error concealment for MPEG video decoder," *SPIE*, vol. 2049, pp. 195–204, 1993.
[18] H. Sun and J. Zdepski, "Adaptive error concealment algorithm for MPEG compressed video," *SPIE*, vol. 1818, pp. 814–824, 1992.
[19] A. Kundu and W. R. Wu, "Wilcoxon–Mann–Whitney rank test-based filter for removing streak noise in transmitted DPCM images," *IEEE Trans. Commun.*, vol. 38, pp. 150–155, Feb. 1990.
[20] C. H. Chang and J. J. Leou, "Detection and correction of transmission errors in DPCM images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 5, pp. 166– 171, Feb. 1995.
[21] C. Y. Chang and J. J. Leou, "Detection and elimination of 2-D transmission error patterns in DPCM images," *IEEE Trans. Commun.*, vol. 44, pp. 1251–1256, Oct. 1996.
[22] W. L. Shyu and J. J. Leou, "Detection and correction of transmission errors in facsimile images," *IEEE Trans. Commun.*, vol. 44, pp. 938–948, Aug. 1996.
[23] L. J. Wang and J. J. Leou, "Detection and correction of transmission errors in pyramid images," in *Proc. 1994 IPPR Conf. Comput. Vision, Graph., Image Processing*, Taiwan, R.O.C., 1994, pp. 304–311.
[24] W. M. Lam and A. R. Reibman, "Self-synchronizing variable-length codes for image transmission," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1992, pp. III-477–III-480.
[25] ——, "An error concealment algorithm for images subject to channel errors," *IEEE Trans. Image Processing*, vol. 4, pp. 533–542, May 1995.
[26] T. P. O'Rourke, R. L. Stevenson, Y. F. Huang, and D. J. Costello, Jr., "Improved decoding of compressed images received over noisy channels," in *Proc. IEEE Int. Conf. Image Processing*, 1995, pp. II-65–II-68.
[27] M. de Prycker, *Asynchronous Transfer Mode: Solution for Broadband ISDN*. New York: Ellis Horwood, 1993.
[28] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Reading, MA: Addison-Wesley, 1992.
[29] Y. H. Han, "Detection and correction of transmission errors in JPEG images," Master thesis, Inst. Comput. Sci. Inform. Eng., Nat. Chung Cheng Univ., Chiayi, Taiwan, R.O.C., 1995.



Yi-Huang Han was born in Pingtung, Taiwan, R.O.C., on April 19, 1971. He received the B.S. degree in computer science and information engineering in 1993 from Fu Jen Catholic University, Taipei, Taiwan, and the M.S. degree in computer science and information engineering in 1995 from National Chung Cheng University, Chiayi, Taiwan.

His current research interests include image processing, pattern recognition, and computer vision.

Jin-Jang Leou was born in Chiayi, Taiwan, R.O.C., on October 25, 1956. He received the B.S. degree in communication engineering in 1979, the M.S. degree in communication engineering in 1981, and the Ph.D. degree in electronics in 1989, all from National Chiao Tung University, Hsinchu, Taiwan.

From 1981 to 1983, he served in the Chinese Army as a Communication Officer. From 1983 to 1984, he was at National Chiao Tung University as a Lecturer. Since August 1989, he has been on the Faculty of the Institute of Computer Science and Information Engineering at National Chung Cheng University, Chiayi, Taiwan. His current research interests include image/video processing, pattern recognition, and computer vision.